LibreHealth

NeoSmartML Foundation Model for Neonatal ICU Monitor Data

Introduction

My name is R Sagar Shresti. I am a final year undergraduate student in Computer Science and Engineering at Angadi Institute of Technology and Management, Belagavi, India. I have a strong interest in Artificial Intelligence, Machine Learning and data driven systems. I enjoy learning how intelligent technologies can be applied to solve real world problems. During my academic journey, I have developed a strong foundation in programming and machine learning concepts. I primarily work with Python and have experience using machine learning and data analysis libraries such as Scikit learn, Pandas and NumPy along with visualization tools like Matplotlib and Seaborn. I am also familiar with deep learning frameworks such as PyTorch, TensorFlow and computer vision tools like OpenCV. I also have experience building web based applications using Flask and managing code through Git and GitHub. In addition to programming and machine learning, I am interested in open source development and collaborative software engineering. I enjoy exploring new technologies, understanding large software systems and contributing to projects that create meaningful impact.

Contact Information

- Name - R Sagar Shresti
- Email - sagar18reshmi@gmail.com
- Github - https://github.com/Sagarshresti18
- GitLab - https://gitlab.com/Sagarshresti18
- Linkedin - https://www.linkedin.com/in/sagar-shresti-976484280
- University - Angadi Institute of Technology and Management
- University Course - Computer Science and Engineering
- Time Zone - Indian Standard Time (IST) +5:30 UTC

Motivation

Why GSoC?

Google Summer of Code offers students meaningful and practical experience within open source projects as well as an opportunity to contribute to the real-world needs of the software community through experience with professional mentors. In addition to allowing me to improve my technical abilities, this experience will also give me the opportunity to develop in a team environment as part of the software development life cycle. I am looking forward to working with the open source community and to gain valuable insight into the process it takes to create and maintain a large, open source project. By participating in GSoC, I believe I will be able to establish myself as a developer and contribute something meaningful to exciting open source projects.

Why LibreHealth?

LibreHealth focuses on developing open-source healthcare solutions that can make a real impact in medical environments. I am particularly interested in healthcare technology because it combines technical innovation with real world benefits for society. Contributing to LibreHealth will allow me to apply my technical skills while working on projects that support healthcare systems and improve patient care.

Why NeoSmartML Foundation Model for Neonatal ICU Monitor Data Project?

The NeoSmartML Foundation Model for Neonatal ICU Monitor Data project is particularly interesting to me because it combines machine learning, computer vision and healthcare monitoring. Developing a system that can analyze neonatal ICU monitor images and extract meaningful clinical insights is both technically challenging and impactful. I believe working on this project will allow me to apply machine learning techniques to a real healthcare problem and contribute to improving neonatal monitoring systems.

Learning Expectations

Through this project, I expect to understand machine learning concepts, computer vision techniques and real world AI applications in healthcare. I also hope to learn best practices in open source development such as collaborative coding, documentation and code reviews while working closely with experienced mentors.

Experience

I have a strong interest in machine learning, data analysis and computer vision. and I have been developing my skills in these areas during my academic journey. I primarily work with Python and have experience using machine learning and data analysis libraries such as Scikit learn, Pandas, NumPy, Matplotlib and Seaborn. I am also familiar with deep learning frameworks such as PyTorch, TensorFlow and computer vision tools like OpenCV. which are commonly used for AI based image analysis systems. In addition to machine learning, I have experience developing web based applications using Flask, Angular and working with databases such as MySQL and MongoDB. I also regularly use Git and GitHub for version control and collaborative development. I have participated in technical events and hackathons, which helped me improve my problem-solving skills and teamwork.

Following are some of the projects I have worked on:

- **Chronic Kidney Disease Detection using Machine Learning** – A machine learning based system that predicts the risk of chronic kidney disease using medical data and achieved 92% accuracy using Random forest algorithm.

- **Driver Drowsiness Detection System** – A computer vision system that detects driver fatigue using facial landmark detection and Eye Aspect Ratio (EAR) to trigger alerts.

LibreHealth Contributions

I have started contributing to the **NeoSmartML repository** in LibreHealth and explored the codebase related to the OCR pipeline used for extracting vital sign values from neonatal ICU monitor images.

Some of my contributions include:

• **MR !21** – Add missing re import to fix NameError in OCR_final.py
I identified a runtime error caused by a missing ’re’ module import in the OCR script used for pattern matching. I fixed the issue by adding the required import to ensure the OCR pipeline runs correctly.

• **MR !20** – Fix hardcoded image path and incorrect filename pattern in OCR pipeline
I fixed an issue where the OCR pipeline used a hardcoded image path that was specific to the original development environment. I modified the implementation to use a more flexible path configuration so that contributors can run the pipeline locally without environment specific issues.

Other Open Source Contributions

I have also contributed to the **scikit-learn open source repository**, one of the widely used machine learning libraries to improve my understanding of collaborative software development and large scale ML codebases.

• **PR #32373 (Merged)**
I updated the code by converting relative imports to absolute imports to improve code clarity and maintainability. The contribution was reviewed by the maintainers and successfully merged into the main scikit-learn repository.

• **Issue #33256 (Resolved / Closed)**
Reported a documentation issue where the BNP Paribas sponsor logo was not rendering correctly in the footer of the scikit-learn documentation website. The issue helped identify a broken image asset and was later closed after the fix was applied.

Project Description

Project Introduction

The project I would like to work on for LibreHealth as part of GSoC 2026 is “NeoSmartML Foundation Model for Neonatal ICU Monitor Data".

The NeoSmartML Foundation Model project aims to develop a comprehensive deep learning model that can extract, analyze and predict various clinical outcomes from neonatal ICU monitor images captured through ESP32-CAM devices. Building upon existing work in fusion models and streaming data analysis, this project extends beyond basic risk classification to create a more versatile foundation model capable of multiple downstream tasks. Look here for earlier architecture: Machine learning-based decision-support.

This application will process real-time monitor images through OCR to extract vital signs data, combining temporal sequences with clinical context to provide various predictions and insights for neonatal care. The foundation model will serve as a base for multiple specialized tasks through fine-tuning or prompt engineering.

Model architecture and Tasks

- The model will utilize a hierarchical architecture combining:
- Vision encoder for raw monitor image processing
- OCR processing layer for text extraction
- Temporal sequence modeling for trend analysis
- Multi-task prediction heads for various clinical outcomes

Target Classification Tasks:

Risk Level Assessment

- Low, moderate, and high-risk classifications
- Continuous risk probability scoring
- Trend-based risk projection

Clinical Event Prediction

- Apnea episodes
- Bradycardia events
- Desaturation events
- Temperature instability

Treatment Response Prediction

- Oxygen therapy effectiveness
- Temperature intervention outcomes
- Feeding tolerance
- Medication response patterns

Physiological State Classification

- Sleep/wake cycles
- Stress levels
- Pain assessment
- Respiratory effort patterns

Deliverables

- Develop a foundation model architecture that handles both image and temporal data
- Create specialized prediction heads for multiple clinical tasks
- Implement efficient model compression for edge deployment
- Build an evaluation framework for model performance
- Create documentation for model usage and fine-tuning
- Develop integration guidelines for clinical applications

Expected Impact

This foundation model will significantly advance the capabilities of neonatal monitoring systems by providing a versatile base model that can be adapted for multiple clinical tasks. The edge-optimized implementation ensures practical deployment in resource-constrained environments while maintaining high accuracy and real-time processing capabilities.

The model's ability to handle multiple tasks through a single foundation architecture will improve efficiency and reduce computational overhead, while the attention-based mechanisms will help identify critical patterns in vital signs data. The project's focus on interpretability and clinical integration ensures that the model's predictions can be effectively utilized in clinical practice.

Proposed Approach

1. Data Collection and Understanding

- Study the existing NeoSmartML dataset and understand the format of neonatal ICU monitor images.
- Identify the vital parameters displayed on the monitor such as heart rate, oxygen saturation (SpO_2), respiratory rate and temperature.

- Understand how images are captured using ESP32-CAM devices and how the data is streamed into the system.
- Analyze the quality of monitor images and identify challenges such as noise, reflections, low lighting or blurred text.

2. Image Preprocessing

- Apply preprocessing techniques to improve the quality of neonatal monitor images before OCR processing.
- Resize and normalize images to maintain consistent input dimensions.
- Use contrast enhancement and noise reduction to improve the visibility of text on the monitor display.
- Use OpenCV based image processing, including cropping and region detection to isolate vital sign display areas for accurate OCR extraction.

3. OCR Based Vital Sign Extraction

- Implement an OCR pipeline to extract numerical values of vital signs from neonatal monitor images.
- Use OCR tools such as Tesseract or similar libraries to detect and recognize text displayed on the monitor.
- Apply regular expressions and pattern matching to identify specific parameters such as heart rate and oxygen saturation.
- Perform post processing to correct OCR errors, validate extracted values and convert them into structured time-series data for further analysis.

4. Time Series Data Processing

- Organize the extracted vital sign values into sequential time series data.
- Handle missing or inconsistent data using preprocessing methods such as interpolation or filtering.
- Analyze temporal patterns in physiological signals such as sudden drops in oxygen saturation or irregular heart rate patterns.
- Prepare the time series data to be used as input for deep learning models.

5. Vision Encoder for Monitor Images

- Implement a vision encoder model to extract meaningful visual features from the monitor images.
- Use convolutional neural networks (CNNs) or lightweight vision models to encode visual information.

- Combine image based features with OCR extracted data to improve model understanding of the monitor display.
- Optimize the encoder architecture to ensure efficient processing for real-time monitoring systems.

6. Temporal Sequence Modeling

- Implement temporal models such as LSTM, GRU or Transformer based architectures to analyze time series patterns in vital signs data.
- Train the model to identify abnormal physiological trends over time.
- Capture relationships between multiple vital parameters such as heart rate and oxygen saturation.
- Use attention mechanisms to help the model focus on important changes in vital signs.

7. Multi Task Prediction Framework

- Design a multi task learning architecture with a shared backbone to support multiple clinical prediction tasks.
- Implement a prediction head for risk level assessment (low, moderate, high risk).
- Implement prediction heads for clinical event detection such as apnea, bradycardia and oxygen desaturation.
- Add prediction modules for treatment response (oxygen therapy or temperature intervention outcomes).
- Include physiological state classification (sleep/wake cycles, stress levels) and train the model to perform all tasks simultaneously using shared features.

8. Model Optimization for Edge Deployment

- Optimize the trained model for deployment on resource constrained edge devices.
- Apply techniques such as model pruning, quantization and lightweight model architectures.
- Reduce computational overhead while maintaining prediction accuracy.
- Ensure the model can perform near real time inference in neonatal monitoring environments.

9. Model Evaluation and Validation

- Evaluate model performance using standard machine learning metrics such as: Accuracy, Precision, Recall, F1-score and AUROC
- Evaluate temporal prediction accuracy for detecting clinical events.
- Test the model under different image conditions such as low lighting or noisy images.
- Compare results with baseline models to measure improvements.

10. Documentation and Integration

- Create clear documentation explaining the model architecture, training process and usage instructions.
- Provide guidelines for fine-tuning the foundation model for different clinical prediction tasks.
- Document the integration with LibreHealth neonatal monitoring systems and the deployment process for clinical environments.

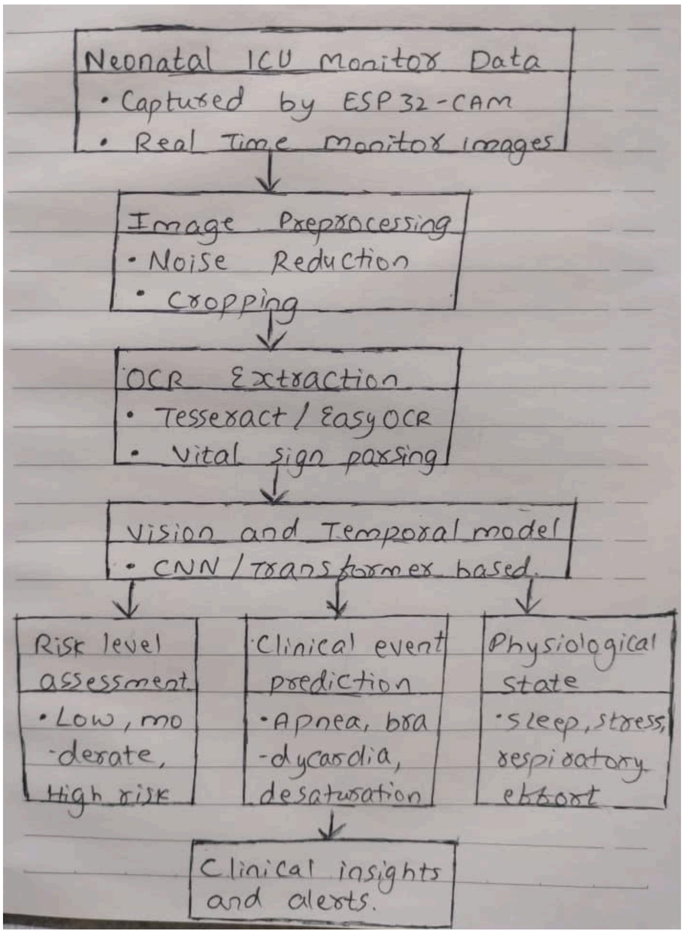

Fig: NeoSmartML System Architecture for Neonatal ICU Monitor Data Analysis

Benefits to the Community

- Improves neonatal patient monitoring by automatically analyzing ICU monitor data and identifying potential health risks.
- Helps medical staff detect critical events early such as apnea, bradycardia and oxygen desaturation.
- Provides an open source AI framework that researchers and developers can extend for other healthcare monitoring tasks.
- Enables real-time analysis on edge devices, making the system practical for hospitals with limited computing resources.
- Supports the LibreHealth ecosystem by adding advanced machine learning capabilities to neonatal monitoring systems.
- Encourages further research and innovation in AI-based healthcare solutions through an open and reusable foundation model.

Preliminary Task

I have completed the preliminary tasks for the NeoSmartML project.

Task 1 – CheXNet Dynamic Quantization
https://gitlab.com/Sagarshresti18/neosmartml-chexnet-quantization
This task demonstrates a clean and reproducible dynamic quantization workflow for a CheXNet-style DenseNet121 backbone using PyTorch. The goal is to reduce model size for deployment-focused scenarios while preserving the architecture and portability of the checkpoint.

Task 2 – Rotation Sensor Motion Detector
https://gitlab.com/Sagarshresti18/neosmartml-rotation-sensor
This Angular app uses the DeviceMotion API to read live rotation data and display device status in real time.

Both repositories include the implementation, documentation and screenshots of the results.

Project Schedule

Timeline	Deliverables
Community Bonding (May 01 - May 24)	Study the NeoSmartML repository, review the OCR pipeline, understand the architecture and discuss implementation strategies with mentors
Week 1-2 (May 25 - June 8)	Image preprocessing pipeline with contrast enhancement, noise reduction, region detection, preprocessing test suite
Week 3-4 (June 8 - June 22)	Enhanced OCR pipeline (Tesseract/EasyOCR), pattern matching for vital signs, error correction, OCR accuracy benchmarks
Week 5-6 (June 22 - July 06)	Vision encoder implementation (ResNet50/EfficientNet), lightweight model testing, feature extraction module, model optimization for edge
Midterm Evaluation (July 06 - July 10)	Midterm Report: OCR pipeline accuracy metrics, vision encoder feature extraction results, integrated image and temporal data preprocessing, preliminary model architecture documentation, code review with mentors, progress assessment against timeline
Week 7-8 (July 10 - July 27)	Temporal sequence modeling (LSTM/Transformer), attention mechanisms, sequence windowing strategy, temporal analysis tests
Week 9 (July 27 - August 10)	Multi task prediction framework with 4 prediction heads like risk assessment, clinical event detection, treatment response, physiological state classification
Week 10 (August 10 - August 17)	Model compression(quantization, pruning), edge deployment optimization, performance benchmarking
Code Submission and Final Evaluation (August 17 - August 24)	Model evaluation, validation, robustness testing, comprehensive documentation, integration guidelines, final code submission

Availability

I will be available to contribute approximately 30–35 hours per week during the Google Summer of Code period. As a final year undergraduate student, I will prioritize my schedule to ensure consistent progress on the project. I do not have any major commitments during the GSoC coding period that would significantly affect my availability. I will maintain regular communication with mentors through GitLab issues, discussions, meetings and I will provide regular progress updates.

In case of any delays or unexpected issues, I will adjust my schedule and dedicate additional time to ensure that project milestones are completed on time.

Why am I a good fit for this project?

Because my technical expertise relates closely to what is required by NeoSmartML, I feel I am a good candidate for this project. I have previously completed many projects using Python, which includes using machine learning libraries, using computer vision tools like OpenCV and developing deep learning models using PyTorch and TensorFlow, thus giving me a lot of experience in building AI based systems for monitoring the health care of individuals.

I have also started to contribute to the NeoSmartML project by fixing problems with OCR pipeline issues in a continuing manner and this has provided me with some understanding of how the project is structured and developed as well as how to extract and gather necessary vital sign information from images of newborns taken with neonatal monitors.

I am interested in open-source projects. I will continue to support the project as much as I can and work on improving the NeoSmartML project. Additionally, I will do my best to learn from my mentors, work with the community and produce a reliable and usable solution during the GSoC program.

Post-GSoC

My goal after the end of Google Summer of Code is to keep contributing to the NeoSmartML project as a part of the LibreHealth ecosystem. I want to improve the foundation model more by refining the OCR pipeline, improving the accuracy of vital signs extraction and making the whole system more robust when dealing with various types of monitor image conditions such as noise, low light or partial obstruction.

Additionally, I will continue to improve on the machine learning models I built during the project through experimentation with different architectures and training strategies. This includes work on the temporal modeling component, improvements in multi task prediction functionality and implementing some optimization strategies in order to develop a model that will be able to perform efficiently on the edge in real time monitored clinical settings.

Another major part of my contribution will be to assist in enhancing the usability and accessibility of NeoSmartML for other developers and researchers. This will include improving the maintenance of the code base by fixing bugs, updating documentation, providing set up instructions and helping new contributors understand the architecture and flow of the project.

As I progress into the future, I intend on being a continuing contributor to the LibreHealth community in terms of involvement and activity. I plan on being involved in both discussions as well as collaborating with developers throughout the development process of open source technology related to healthcare. Thus, by continuing my involvement with LibreHealth after the conclusion of GSoC, I want to support the NeoSmartML project and help create an effective and viable platform for AI-based monitoring of neonates and healthcare research.

References:

- LibreHealth NeoSmartML Project Repository
  https://gitlab.com/librehealth/incubating-projects/mhbs/neosmartml
- Tesseract OCR Documentation
  https://tesseract-ocr.github.io/
- OpenCV Documentation
  OpenCV: OpenCV modules
- PyTorch Documentation
  https://pytorch.org/docs/
- TensorFlow Documentation
  https://www.tensorflow.org/
- Scikit-learn Machine Learning Library
  https://scikit-learn.org/
- PhysioNet – Clinical Physiological Data Resources
  https://physionet.org/